



Comunicação em rede

Comunicação em rede no processo de desenvolvimento de jogos digitais

A rede tornou-se o principal ponto de venda para jogos de computador. É esperado que os projetos comerciais suportem os jogos multiplayer, e os sites têm como objetivo oferecer suporte a um número cada vez maior de usuários. Ao mesmo tempo, os lançamentos de novos consoles dependem fortemente do apelo dos projetos online, e um novo ramo de entretenimento móvel surgiu com a intenção de desenvolver jogos multijogador distribuídos para aplicativos sem fio.

Os jogos de computador multijogador exigem uma rede consistente e responsiva. A consistência é importante para manter um conjunto de dados semelhante para todos os jogadores, enquanto a capacidade de resposta exige que as atualizações dos dados sejam feitas o mais rápido possível. Esses dois requisitos, no entanto, são muitas vezes contraditórios e resolver essa dicotomia consistência-responsividade está no cerne da rede interativa em tempo real (ARRUDA, 2014).

Um jogo multiplayer massivo pode ter dezenas de milhares de jogadores simultâneos de todo o mundo, o que significa que a escalabilidade da arquitetura de rede escolhida torna-se crítica. Além disso, geralmente requer a manutenção de um mundo persistente, onde o jogo progride o tempo todo, independentemente de um jogador participar ou não dele.

Já os jogos móveis e sem fio requerem uma atenção especial para manter um fluxo de informações contínuo e livre de erros. Os sites visam fornecer aos jogadores um conteúdo mais personalizado conforme a sua preferência e estilo. Isso requer métodos para analisar o comportamento

do jogador, o que também é importante para detectar jogadores que trapaceiam ou se comportam mal no mundo.



Os microsserviços tornam seu jogo mais gerenciável

Uma arquitetura orientada a microsserviços divide um grande aplicativo, neste caso, o jogo em pequenos serviços modulares com versões independentes que se comunicam entre si por meio de APIs simples e universalmente acessíveis. Isso torna muito mais fácil criar funcionalidades e mantê-las após a sua criação.

Separar os servidores de jogo da funcionalidade de bate-papo torna toda a infraestrutura mais gerenciável e aproxima o projeto de uma arquitetura totalmente orientada a microsserviços. Com uma arquitetura monolítica, a equipe de desenvolvimento está presa a uma única pilha de tecnologia — usando as mesmas linguagens de programação, bancos de dados e ambientes de software nos quais o jogo já foi criado. Ao trazer novos desenvolvedores ou quando desejamos prototipar novas tecnologias e sistemas, é muito mais fácil mover-se rapidamente em uma arquitetura de microsserviços (ALBUQUERQUE, 2019).

As dependências também são muito mais aparentes com arquiteturas monolíticas. Se a função de um único aplicativo falhar, todo o jogo ficará inoperante. Dividir o jogo em microsserviços facilita isolar uma falha e corrigi-la se um único módulo falhar. Os servidores de jogo são construídos para fornecer movimento e estado do jogador em tempo real, e eles fazem isso muito bem. Adaptar a mesma tecnologia e design para mensagens de bate-papo simplesmente não representa a utilização das melhores opções para a funcionalidade específica. Componentes descentralizados são mais fáceis de manter e escalam melhor. Além disso, também podemos executar outros serviços fora dos servidores do

jogo, incluindo autorização, presença, estatísticas e tabelas de classificação.



Desempenho de bate-papo de um jogo digital

No geral, o desempenho é uma consideração importante para um jogo multiplayer. Uma experiência lenta afastará os usuários e eles nunca mais retornarão. Com uma arquitetura monolítica, o projeto pode funcionar em um laboratório. Porém, para jogos multijogador com muitos usuários localizados em qualquer lugar do mundo, todos se comunicando simultaneamente em um ritmo rápido, você começará a ver atrasos e latências na entrega das mensagens de bate-papo e na experiência.

A separação dos dois garante que os recursos da CPU e da rede sejam usados com mais eficiência. O objetivo principal dos servidores é oferecer uma experiência perfeita para todos os usuários. Como resultado, o poder de processamento deve ser usado para maximizar esse desempenho. Por exemplo, vamos considerar que você tenha um jogo de arena de batalha online como *League of Legends*, no qual podemos ter centenas de jogadores em um único mundo simultaneamente. São milhares de mensagens enviadas pelos servidores, fornecendo todas as entradas que cada jogador cria. Agora, adicione mensagens de bate-papo nesse processo. É totalmente possível que os jogadores enviem spam para o canal de bate-papo e desacelerem deliberadamente o servidor, pois todas as mensagens teriam a mesma prioridade. Claro, seria possível verificar esses usuários, mas você exigiria processamento adicional, o que consumiria os recursos do servidor. O servidor já está lidando com experiências intensivas — física, gráficos e som. Quando você adiciona mensagens de bate-papo - individuais, em grupo ou em equipe - e analisa e encaminha as mensagens para os usuários corretos, todas

essas mensagens se acumulam lentamente em grande escala e prejudicam o desempenho geral do jogo (RABIN, 2012).



Portanto, não é prudente executar os canais de bate-papo separadamente dos canais multijogador. Essa dinâmica está roubando um poder de processamento importante que poderia ser mais adequado para problemas mais complexos do que o roteamento de mensagens de bate-papo.

Atividade Extra

A partir do que foi apresentado neste módulo, principalmente no aprofundamento dos princípios de comunicação em rede no processo de desenvolvimento de jogos digitais, recomenda-se a leitura do artigo científico "Desenvolvimento de um jogo digital sobre redes de computadores" (SANTOS et. al., 2019), disponível no Google Acadêmico e no diretório da Universidade Federal Fluminense.

Será possível observar o processo de criação de um jogo digital sobre redes de computadores, cujo objetivo é proporcionar diversão e, como consequência, fazer com que o jogador adquira conhecimento. O ambiente do jogo é didático, no qual as escolhas do jogador podem resultar em falhas e, para avançar no jogo, é essencial aprender a fazer as escolhas corretas.

O jogo apresentado nesta pesquisa relaciona alguns conceitos sobre as camadas da pilha de protocolos da Internet e também mostra como funcionam alguns serviços, como os protocolos DHCP e DNS da camada de aplicação, transmissão confiável, controle de fluxo e integridade da camada de transporte, broadcast e roteamento da camada de rede, tabela ARP e multiacesso da camada de enlace.



Referência Bibliográfica

ARRUDA, E. P. **Fundamentos para o Desenvolvimento de Jogos Digitais**. Porto Alegre: Grupo A, 2014.

ALBUQUERQUE, R. M. **Estudo contemporâneos em design de jogos e entretenimento digital**. Porto Alegre: SAGAH, 2019.

RABIN, S. **Introdução ao Desenvolvimento de Games**. São Paulo: Cengage Learning Brasil, 2012.

Ir para exercício